

OPTIMASI PENJADWALAN PERKULIAHAN MENGGUNAKAN METODE *AUTO GENERATE TIMETABLE* DENGAN ARRAY

Hani Dewi Ariessanti¹

Dwi Sloria Suharti²⁾

Ary Budi Warsito³⁾

Dosen STMIK Raharja Jurusan Sistem Komputer¹, Universitas Muhammadiyah Tangerang², Dosen
STMIK Raharja Jurusan Teknik Informatika³

Email : hani.dewi@raharja.info, dsloria@gmail.com, arybudiwarsito@raharja.info

Diterima: 4 Juli 2018/ Disetujui : 20 Juli 2018

ABSTRACT

Scheduling is one of the many problems that has been done in research for many years. Problems preparing the schedule in college are about college scheduling. The timing process should be done for each semester, which is a tiring and time-consuming task. The overall allocation of events in timeslots and spaces is done by the course scheduling process by considering the list of hard constraints and soft constraints, so there is no conflict created in allocating the schedule. Therefore, it is necessary to create a lecture scheduling application that is able to facilitate and overcome the problems in organizing the lecture schedule. The proposed scheduling system design proposed in this study is to optimize the schedule of lectures using the Auto Generate Timetable method with arrays to find the best candidates for college scheduling with the aim of minimizing the conflict and optimizing the scheduling schedule. This method is based on the process of lecturing process that has been conducted in college. Every curriculum, space, day / time, is needed to arrange the schedule of students and lecturers as part of the scheduling variable that is the solution candidate. Then the process of adjusting to constraints has been made with various parameters. The research method is to collect the data, analysis, design, coding, testing and up to the maintenance phase by using the Waterfall System Development Lifecycle method. The waterfall model provides a life-cycle approach to the development of software systems in sequential or sequential form. So with the existence of this application, it is hoped that the arrangement of lectures will not find problems as a constraint in arranging the lecture schedule.

Keywords: Auto Generate Timetable, Array, constraint, waterfall, SDLC

ABSTRAK

*Penjadwalan adalah salah satu masalah yang telah banyak dilakukan dalam penelitian selama bertahun-tahun. Masalah penyusunan jadwal di perguruan tinggi adalah seputar penjadwalan kuliah. Proses penentuan waktu harus dilakukan untuk setiap semester, yang merupakan tugas yang melelahkan dan memakan waktu. Alokasi keseluruhan kejadian di timeslots dan ruang dilakukan oleh proses penentuan penjadwalan kuliah dengan mempertimbangkan daftar hard constraint dan soft constraint, sehingga tidak ada konflik yang tercipta dalam mengalokasi jadwal tersebut. Oleh karena itu, perlu dibuat sebuah aplikasi penjadwalan perkuliahan yang mampu memudahkan dan mampu menanggulangi permasalahan dalam pengaturan jadwal perkuliahan. Rancangan sistem penjadwalan perkuliahan yang diusulkan dalam penelitian ini bertujuan untuk mengoptimalkan penyusunan jadwal perkuliahan menggunakan metode *Auto Generate Timetable* dengan array untuk mencari kandidat terbaik penjadwalan kuliah dengan tujuan meminimalisir konflik dan mengoptimalkan penyusunan jadwal. Metode ini dilakukan berdasarkan pada mekanisme proses perkuliahan yang telah berjalan pada perguruan tinggi. Setiap*

kurikulum, ruang, hari/waktu, dibutuhkan untuk menyusun jadwal mahasiswa dan dosen sebagai bagian dari variabel penjadwalan yang merupakan kandidat solusi. Kemudian dilakukan proses penyesuaian terhadap constraint-constraint yang telah dibuat dengan berbagai parameter. Metode penelitian yang dilakukan adalah dengan mengumpulkan data, analisis, desain, pengkodean, pengujian dan sampai tahap pemeliharaan dengan menggunakan metode Siklus Hidup Pengembangan Sistem jenis waterfall. Model waterfall menyediakan pendekatan siklus hidup pengembangan sistem perangkat lunak dalam bentuk sekuensial atau terurut. Sehingga dengan adanya aplikasi ini, diharapkan pengaturan perkuliahan tidak menemukan permasalahan sebagai kendala dalam menyusun jadwal perkuliahan.

Kata Kunci: *Auto Generate Timetable, Array, constraint, waterfall, SDLC*

1. PENDAHULUAN

Penjadwalan perkuliahan merupakan salah satu komponen penting dalam suksesnya pelaksanaan kegiatan belajar mengajar di perguruan tinggi, karena kegiatan penjadwalan dosen dan penjadwalan kuliah mahasiswa bergantung pada jadwal yang ada sehingga jadwal harus disusun dengan benar pada setiap semester. Pembuatan jadwal kuliah adalah kegiatan rutin yang dilakukan oleh perguruan tinggi pada tiap tahun ajaran. Di perguruan tinggi, program penjadwalan merupakan sesuatu yang penting dalam proses belajar mengajar, karena semua kegiatan dosen dan kegiatan mahasiswa bergantung pada jadwal yang ada, sehingga harus disusun dengan benar dan diperbaiki pada awal tahun akademik, sehingga nantinya tidak mengganggu aktifitas belajar mengajar antar dosen dan mahasiswa (Ariani, Fahriza, Prasetyaningrum, 2011). Terdapat beberapa aspek yang menjadi faktor penting adalah menyusun jadwal perkuliahan diantaranya adalah tidak boleh adanya jadwal yang konflik misalnya ada seorang dosen yang jadwalkan mengajar di 2 kelas jelas berbeda pada saat yang bersamaan atau mahasiswa dijadwalkan di 2 matakuliah yang pada jam yang sama, ketersediaan waktu mengajar bagi dosen tidak tetap, keterbatasan ruangan, distribusi mata kuliah yang merata disetiap minggunya. Selain itu kendala yang terjadi adalah pada penyusunan penjadwalan kuliah masalah dari segi mahasiswa, keterbatasan jumlah dosen untuk mengajar kuliah. Selain itu ketersediaan kelas hingga kegiatan belajar yang dilaksanakan yang ini menjadi permasalahan diatas biasanya dikenal dengan University Timetabling Problem (UTP).

Berdasarkan beberapa aspek diatas maka diperlukan proses pembuatan jadwal tidak terlalu banyak atau meminimalkan *constraint* maka sehingga diperlukan optimasi yang dapat diterapkan dalam membuat penjadwalan kuliah. Optimasi ini tidak bisa sepenuhnya menghasilkan hasil yang optimal tapi mendekati optimal. Ada beberapa metode optimasi yang sering digunakan untuk menyelesaikan penjadwalan dengan keunggulan masing-masing. (Aziz, 2015) Metode tersebut antara lain: Ant Colony, Simulated Anneling, Genetic Algorithm. Adapun kelebihan Ant Colony adalah dapat diterapkan secara sempurna dengan perubahan minimal ke permasalahan optimasi kombinasi, Simulated Anneling memiliki keunggulan lebih cepat dalam menyelesaikan iterasinya sedangkan Genetic Algorithm dapat digunakan untuk mencari berbagai solusi persoalan yang ada di dunia nyata. Selain itu beberapa penelitian sebelumnya telah dilakukan komparasi antara Algoritma Genetika dengan Algoritma Palgunadi untuk penjadwalan mata kuliah (Yudihartanti, Syukur, & Wahono, 2011). Penelitian ini telah menghasilkan sebuah kombinasi algoritma genetika dan algoritma palgunadi dengan dihasilkannya perbaikan proses heuristik dari Algoritma Genetika dibuktikan dengan semakin baiknya nilai fitnessnya yang dihasilkan dan semakin sedikit waktu proses yang dibutuhkan. (Siswono, 2013). Selain itu telah

dilakukan analisa terhadap perbandingan Metode Mamdani dan Metode Sugeno dalam penjadwalan mata kuliah dimana terdapat perbedaan pada proses yang dilakukan misalnya pada proses implikasi agregasi dan proses defuzzifikasinya. Pada proses ujicoba dengan dataset didapatkan bahwa hasil komparasi menunjukkan bahwa metode Mamdani mempunyai tingkat ketepatan yang dibandingkan dengan metode sugeno (Yudihartanti et al., 2011).

Pada penelitian ini diusulkan menggunakan Auto Generate timetable dengan Array Penggunaan Auto Generate timetable merupakan metode optimasi yang digunakan untuk penjadwalan perkuliahan untuk memecahkan masalah optimasi multidimensi dengan waktu proses yang lebih cepat. Metode penelitian yang dilakukan adalah dengan mengumpulkan data, analisis, desain, pengkodean, pengujian dan sampai tahap pemeliharaan dengan menggunakan metode Siklus Hidup Pengembangan Sistem atau jenis waterfall. Model waterfall menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut. Sehingga dengan adanya aplikasi ini, diharapkan pengaturan perkuliahan tidak menemukan permasalahan sebagai kendala dalam menyusun jadwal perkuliahan.

2. METODE PENELITIAN

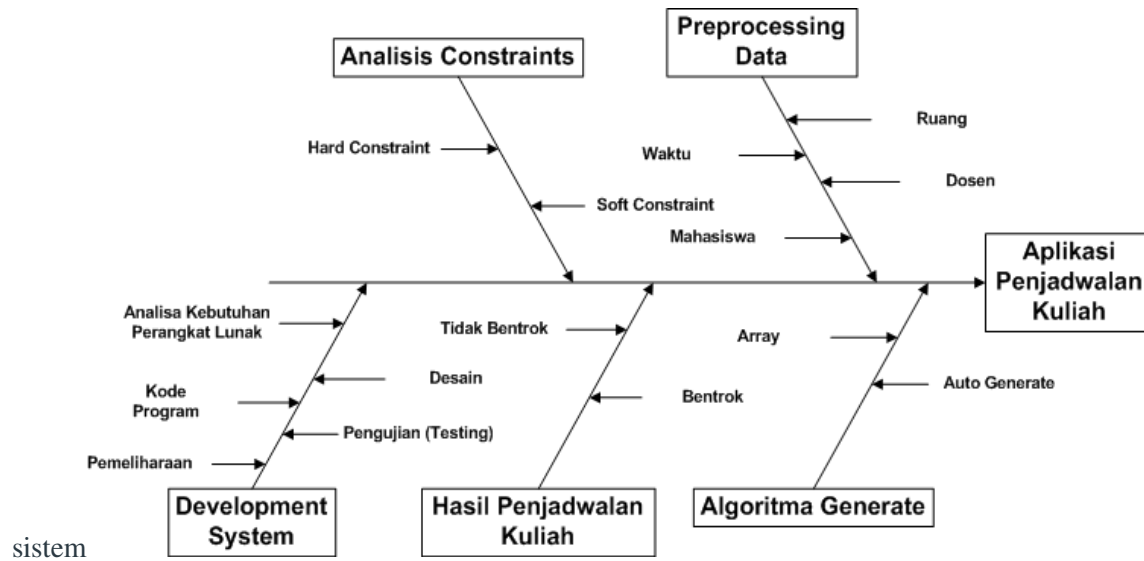
Untuk membangun Aplikasi penjadwalan mata kuliah dengan metode auto generate timetable dengan array, metode penelitian yang digunakan meliputi Metode Pengumpulan Data, Metode Pengembangan Sistem dan pengembangan antar muka pengguna (user interface).

Metode Pengumpulan Data

Tahap pertama peneliti melaksanakan observasi (pengamatan) di salah satu kampus di Tangerang dengan cara mengumpulkan data, informasi dan mempelajari catatan serta dokumen yang ada mengenai prosedur penjadwalan perkuliahan. Selain pengamatan langsung dilakukan teknik pengumpulan data melalui wawancara mendalam (indept interview), Focus Group Discussion (FGD) dan studi dokumenter.

Berdasarkan data pertama ini akan dirancang (desain) dan diimplementasikan sistem penjadwalan perkuliahan menggunakan metode auto generate timetable dengan array yang diujicobakan di salah satu kampus di Tangerang sebagai objek penelitiannya. Pada tahap kedua peneliti melaksanakan kegiatan kebutuhan untuk untuk penjadwalan perkuliahan seperti hard constraint yang berhubungan dengan kepuasan dalam menghasilkan jadwal yang fisibel, soft constraint yang berhubungan dengan kualitas sistem penjadwalan, dan parameter seperti: ruang, waktu, dosen, mahasiswa, kurikulum.

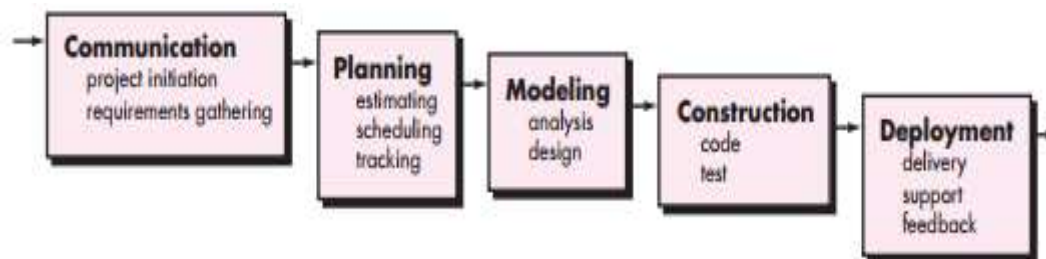
Strategi kegiatan penelitian ini akan dilakukan dengan mekanisme seperti terlihat dalam fishbone diagram berikut ini: Gambar 1. Diagram Fishbone Penelitian Metode Pengembangan Sistem Metode penelitian yang digunakan adalah metode Siklus Hidup Pengembangan jenis Waterfall yang cocok untuk menggambarkan sistem.



Gambar 1. Diagram Fishbone Penelitian

Metode Pengembangan Sistem

Metode penelitian yang digunakan adalah metode SDLC (*System Development Life Cycle*) jenis *Waterfall* yang cocok untuk menggambarkan sistem. Menurut Pressman (Pressman, 2010), model Waterfall atau sering disebut model sekuensial linier atau alur hidup klasik, menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari Communication, Planning, Modeling, Construction, dan Deployment seperti yang disajikan pada gambar 3.2



Gambar 2. Model Waterfall

Langkah-langkah model *Waterfall* adalah sebagai berikut:

1. *Communication*, terdiri dari: *Project Initiation* dan *Requirements Gathering*
2. *Planning*, terdiri dari: *estimating*, *scheduling*, *tracking*.
3. *Modeling*, terdiri dari: *Analysis* dan *Design*
4. *Construction*, terdiri dari: *Code* dan *Test*.
5. *Deployment*, terdiri dari: *Delivery*, *Support* dan *Feedback*.

Prototipe Antarmuka Pengguna

Antarmuka sistem dirancang terdiri dari beberapa form antarmuka yang berfungsi untuk antarmuka pengisian data. Rancangan antarmuka aplikasi ini meliputi:

1. Rancangan form Pengolahan Data Mata Kuliah

Dalam Rancangan ini berisikan informasi Kode, nama mata kuliah dan jumlah SKS yang akan diambil. Dimana dalam form tersebut disediakan menu edit yang dapat digunakan oleh *user* atau pengguna.

Data Mata Kuliah		
Kode	: XX-YYYYY	
Mata Kuliah	: XXXXXXXXXXXX	
SKS	: X	
Kode MK XX-YYYYY	Mata Kuliah XXXXXXXXXXXXXX	SKS X
Tamb	Edit	Hapus
Simpa	Batal	Tutup

Gambar 3. Antarmuka Pengolahan Data Mata Kuliah

2. Rancangan form Pemasukan Data Ruangan

Dalam Rancangan ini berisikan informasi Kode, nama ruangan dan keterangan. Dimana dalam form tersebut disediakan menu edit yang dapat digunakan oleh *user* atau pengguna.

Data Ruangan		
Kode Ruangan	: XXXX	
Nama Ruangan	: XXXXXXXXX	
Keterangan	: XXXXXXXXXXXXX	
Kode Ruangan	Nama Ruangan	Keterangan
XXXX	XXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXX
Tamb	Edit	Hapu
Simp	Batal	Tutup

Gambar 4. Antarmuka Pemasukan Data Ruangan

3. Rancangan form Pengolahan Data Set Mata Kuliah

Dalam Rancangan ini berisikan informasi Kode mata kuliah, kode dosen, durasi pelaksanaan mata kuliah dan keterangan. Dimana dalam form tersebut disediakan menu edit yang dapat digunakan oleh *user* atau pengguna.

Kode Mata Kuliah : XX-YYYY Kode Dosen : XYYYY Waktu Per SKS : XX Keterangan : XXXXXXXXXXXXXXXXXXXX						
Kode MK	Mata Kuliah	SKS	Waktu/SKS	Waktu	Nama Dosen	Keterangan
XX-YYYY	XXXXXXXX	X	XX	XXX	XXXXXXXXXX	XXXXXXXXXXXX
<div> <input type="button" value="Tamba"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/> <input type="button" value="Simpan"/> <input type="button" value="Batal"/> <input type="button" value="Tutup"/> </div>						

Gambar 5. Antarmuka Pengolahan Data Set Mata Kuliah

4. Rancangan form Pengolahan Data Set Waktu kuliah

Dalam Rancangan ini berisikan informasi hari, jam mulai dan selesai pelaksanaan mata kuliah juga keterangan. Dimana dalam form tersebut disediakan menu edit yang dapat digunakan oleh *user* atau pengguna.

Data Set Waktu Kuliah			
Hari : XXXXXX Jam Mulai : XX : YY : ZZ Jam Selesai : XX : YY : ZZ Keterangan : XXXXXXXXXXXXXXXXXXXX			
Hari	Jam Mulai	Jam Selesai	Keterangan
XXXXXX	XX : XX : XX	XX : XX : XX	XXXXXXXXXXXXXXXXXXXX
<div> <input type="button" value="Tambah"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/> <input type="button" value="Simpan"/> <input type="button" value="Batal"/> <input type="button" value="Tutup"/> </div>			

Gambar 6. Antarmuka Pengolahan Data Set Waktu Kuliah

5. Rancangan form *soft Constraint* Waktu Kuliah

SENIN [Jam Ke]	SELASA [Jam Ke]	RABU [Jam Ke]	KAMIS [Jam Ke]	JUMAT [Jam Ke]	SABTU [Jam Ke]
<input type="checkbox"/> 08:00 - 08:50 [1]	<input type="checkbox"/> 08:00 - 08:50 [1]	<input type="checkbox"/> 08:00 - 08:50 [1]	<input type="checkbox"/> 08:00 - 08:50 [1]	<input type="checkbox"/> 08:00 - 08:50 [1]	<input type="checkbox"/> 08:00 - 08:50 [1]
<input type="checkbox"/> 08:50 - 09:40 [2]	<input type="checkbox"/> 08:50 - 09:40 [2]	<input type="checkbox"/> 08:50 - 09:40 [2]	<input type="checkbox"/> 08:50 - 09:40 [2]	<input type="checkbox"/> 08:50 - 09:40 [2]	<input type="checkbox"/> 08:50 - 09:40 [2]
<input type="checkbox"/> 09:40 - 10:30 [3]	<input type="checkbox"/> 09:40 - 10:30 [3]	<input type="checkbox"/> 09:40 - 10:30 [3]	<input type="checkbox"/> 09:40 - 10:30 [3]	<input type="checkbox"/> 09:40 - 10:30 [3]	<input type="checkbox"/> 09:40 - 10:30 [3]
<input type="checkbox"/> 10:30 - 11:20 [4]	<input type="checkbox"/> 10:30 - 11:20 [4]	<input type="checkbox"/> 10:30 - 11:20 [4]	<input type="checkbox"/> 10:30 - 11:20 [4]	<input type="checkbox"/> 10:30 - 11:20 [4]	<input type="checkbox"/> 10:30 - 11:20 [4]
<input type="checkbox"/> 11:20 - 12:10 [5]	<input type="checkbox"/> 11:20 - 12:10 [5]	<input type="checkbox"/> 11:20 - 12:10 [5]	<input type="checkbox"/> 11:20 - 12:10 [5]	<input type="checkbox"/> 11:20 - 12:10 [5]	<input type="checkbox"/> 11:20 - 12:10 [5]
<input type="checkbox"/> 12:10 - 13:00 [6]	<input type="checkbox"/> 12:10 - 13:00 [6]	<input type="checkbox"/> 12:10 - 13:00 [6]	<input type="checkbox"/> 12:10 - 13:00 [6]	<input type="checkbox"/> 12:10 - 13:00 [6]	<input type="checkbox"/> 12:10 - 13:00 [6]
<input type="checkbox"/> 13:00 - 13:50 [7]	<input type="checkbox"/> 13:00 - 13:50 [7]	<input type="checkbox"/> 13:00 - 13:50 [7]	<input type="checkbox"/> 13:00 - 13:50 [7]	<input type="checkbox"/> 13:00 - 13:50 [7]	<input type="checkbox"/> 13:00 - 13:50 [7]
<input type="checkbox"/> 13:50 - 14:40 [8]	<input type="checkbox"/> 13:50 - 14:40 [8]	<input type="checkbox"/> 13:50 - 14:40 [8]	<input type="checkbox"/> 13:50 - 14:40 [8]	<input type="checkbox"/> 13:50 - 14:40 [8]	<input type="checkbox"/> 13:50 - 14:40 [8]
<input type="checkbox"/> 14:40 - 15:30 [9]	<input type="checkbox"/> 14:40 - 15:30 [9]	<input type="checkbox"/> 14:40 - 15:30 [9]	<input type="checkbox"/> 14:40 - 15:30 [9]	<input type="checkbox"/> 14:40 - 15:30 [9]	<input type="checkbox"/> 14:40 - 15:30 [9]
<input type="checkbox"/> 15:30 - 16:20 [10]	<input type="checkbox"/> 15:30 - 16:20 [10]	<input type="checkbox"/> 15:30 - 16:20 [10]	<input type="checkbox"/> 15:30 - 16:20 [10]	<input type="checkbox"/> 15:30 - 16:20 [10]	<input type="checkbox"/> 15:30 - 16:20 [10]
<input type="checkbox"/> 16:20 - 17:10 [11]	<input type="checkbox"/> 16:20 - 17:10 [11]	<input type="checkbox"/> 16:20 - 17:10 [11]	<input type="checkbox"/> 16:20 - 17:10 [11]	<input type="checkbox"/> 16:20 - 17:10 [11]	<input type="checkbox"/> 16:20 - 17:10 [11]
<input type="checkbox"/> 17:10 - 18:00 [12]	<input type="checkbox"/> 17:10 - 18:00 [12]	<input type="checkbox"/> 17:10 - 18:00 [12]	<input type="checkbox"/> 17:10 - 18:00 [12]	<input type="checkbox"/> 17:10 - 18:00 [12]	<input type="checkbox"/> 17:10 - 18:00 [12]
<input type="checkbox"/> 18:30 - 19:20 [13]	<input type="checkbox"/> 18:30 - 19:20 [13]	<input type="checkbox"/> 18:30 - 19:20 [13]	<input type="checkbox"/> 18:30 - 19:20 [13]	<input type="checkbox"/> 18:30 - 19:20 [13]	<input type="checkbox"/> 18:30 - 19:20 [13]
<input type="checkbox"/> 19:20 - 20:10 [14]	<input type="checkbox"/> 19:20 - 20:10 [14]	<input type="checkbox"/> 19:20 - 20:10 [14]	<input type="checkbox"/> 19:20 - 20:10 [14]	<input type="checkbox"/> 19:20 - 20:10 [14]	<input type="checkbox"/> 19:20 - 20:10 [14]
<input type="checkbox"/> 20:10 - 21:00 [15]	<input type="checkbox"/> 20:10 - 21:00 [15]	<input type="checkbox"/> 20:10 - 21:00 [15]	<input type="checkbox"/> 20:10 - 21:00 [15]	<input type="checkbox"/> 20:10 - 21:00 [15]	<input type="checkbox"/> 20:10 - 21:00 [15]
<input type="checkbox"/> All	<input type="checkbox"/> All	<input type="checkbox"/> All	<input type="checkbox"/> All	<input type="checkbox"/> All	<input type="checkbox"/> All

Gambar 7. Antarmuka *Soft Constraints* Waktu Kuliah6. Rancangan *Form Hasil Generate* (Pembangkitan) Penggunaan Ruang Kelas Terhadap Waktu (*Timeslot*)

Senin																								
Jam	L-310	L-211	L-212	L-213	L-303	L-304	L-305	L-308	L-307	L-309	L-210	L-209	L-208	L-205	L-204	L-203	M-104	M-203	M-105	M-103	M-102	M-107	M-204	
08:00 - 08:50	ML301A	-	-	-	SI221A	-	-	-	-	IL191D	MT183B	DL301A	SI152D	BL101D	TI500B	TI102A	PL102A	UL111A	-	-	-	-	-	
08:50 - 09:40	ML301A	-	-	-	SI221A	-	-	-	-	IL191D	MT183B	DL301A	SI152D	BL101D	TI500B	TI102A	PL102A	UL111A	-	-	-	-	-	
09:40 - 10:30	-	-	-	-	TI201A	-	-	-	-	UL601D	PR203D	PL301A	SI103C	ML100A	AK106A	TI102AP	PL102C	UL101C	-	-	SI221AP	BL101P	-	
10:30 - 11:20	-	-	-	-	TI201A	-	-	-	-	UL601D	PR203D	PL301A	SI103C	ML100A	AK106A	TI172A	PL102C	UL101C	-	-	-	-	-	
11:20 - 12:10	-	-	-	-	TI201AP	-	-	-	-	IL201A	PR183B	SI103A	SI103CP	PA102D	AK106AP	TI172A	BL101A	EL104B	-	-	PR203DP	-	-	
12:10 - 13:00	-	-	-	-	SI283C	-	-	-	-	IL201A	PR183B	SI103A	-	PA102D	SI103A	SI330B	BL101A	EL104B	-	-	-	-	-	
13:00 - 13:50	-	-	-	-	SI283C	-	-	-	-	VL301C	PD100A	VL401A	MT103B	PA113C	SI103A	SI330B	PL101A	TL101A	-	-	PR183BP	-	-	
13:50 - 14:40	-	-	-	-	TI110A	-	-	-	-	VL301C	PD100A	VL401A	MT103B	PA113C	SI103AP	SI330A	PL101A	TL101A	-	-	SI283CP	-	-	
14:40 - 15:30	-	-	-	-	TI110A	-	-	-	-	VL201A	PD100AP	SI401C	PA102B	MT103E	SI103D	SI330A	PL101B	ML202A	-	-	PA113CP	-	-	
15:30 - 16:20	GL301A	-	-	-	SI360B	-	-	-	-	VL301A	PR204B	SI401C	PA102B	MT103E	SI103D	SI152F	PL101B	ML202A	-	-	-	-	-	
16:20 - 17:10	GL301A	-	-	-	SI360B	-	-	-	-	-	PR204B	-	-	-	SI103DP	SI152F	-	-	-	-	-	-	-	
17:10 - 18:00	PL601C	-	-	-	TI102D	SI103L	MT112F	PR183H	AK133D	VL401D	BL101I	-	BL101K	LA103J	PA115G	PA102G	PD100BP	-	-	PR203FP	AK106BP	PA115JP	BL101BP	
18:00 - 19:20	PL601C	-	-	-	TI102D	SI103L	MT112F	PR183H	AK133D	VL401D	BL101I	BL101D	BL101K	LA103J	PA115G	PA102G	UL101E	ML101E	SI283CP	MT131EP	LA103JP	BL101JP	-	
19:20 - 20:10	VL301D	SI283D	-	-	AK106B	MT151E	MT112I	PR203F	PD100B	ML202D	BL101J	BL101D	BL101G	LA103I	PA115J	PA102H	UL101E	ML101E	PR183HP	TI102DP	PA115GP	BL101GP	-	
20:10 - 21:00	VL301D	SI283D	-	-	AK106B	MT151E	MT112I	PR203F	PD100B	ML202D	BL101J	-	BL101G	LA103I	PA115J	PA102H	-	-	AK133CP	SI103LP	LA103JP	BL101JP	-	
Sabtu																								
Jam	L-310	L-211	L-212	L-213	L-303	L-304	L-305	L-308	L-307	L-309	L-210	L-209	L-208	L-205	L-204	L-203	M-104	M-203	M-105	M-103	M-102	M-107	M-204	
08:00 - 08:50	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
08:50 - 09:40	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
09:40 - 10:30	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
10:30 - 11:20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
11:20 - 12:10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
12:10 - 13:00	SK140F	-	-	-	-	-	-	SI139E	MT185F	MT185D	MT183C	PR183A	SI140A	AK133A	PR201E	MT103H	MT103I	-	-	-	-	-	-	
13:00 - 13:50	SK140F	-	-	-	-	-	-	SI139E	MT185F	MT185D	MT183C	PR183A	SI140A	AK133A	PR201E	MT103H	MT103I	-	-	-	-	-	-	
13:50 - 14:40	SI360E	-	-	-	-	-	-	PR204D	SI139D	TI143D	PR204C	SI142A	SI340AP	PR183C	SI103F	PA102I	MT103F	-	-	PR183AP	AK133AP	PR201EP	-	
14:40 - 15:30	SI360E	-	-	-	-	-	-	PR204D	SI139D	TI143D	PR204C	SI142A	SI340B	PR183C	SI103F	PA102I	MT103F	-	-	-	-	-	-	
15:30 - 16:20	SI360C	-	-	-	-	-	-	MT183D	TI143C	MT185E	SI360D	SI142AP	SI340B	SI139A	SI103FP	MT103J	UM110G	-	-	-	-	PR183CP	-	
16:20 - 17:10	SI360C	-	-	-	-	-	-	MT183D	TI143C	MT185E	SI360D	-	SI340BP	SI139A	-	MT103J	UM110G	-	-	-	-	-	-	
17:10 - 18:00	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
18:00 - 19:20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
19:20 - 20:10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
20:10 - 21:00	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	

Gambar 8. Antarmuka Hasil Generate (Pembangkitan) Penggunaan Ruang Kelas terhadap waktu (Timeslot)

3. PEMBAHASAN

Timetable

Timetable merupakan alokasi subjek yang memiliki kendala untuk ditempatkan pada ruang waktu (Gani, 2004) permasalahan pada *timetable* cukup luas. Masalah ini ada pada keseharian kehidupan, pendidikan, kesehatan, transportasi, olahraga dan perusahaan produksi. (Chu & Chen, 2006). Proses penyelesaian *timetable* yang optimal cukup rumit dan memakan waktu jika diselesaikan secara manual (Norberciak, 2008). Dari permasalahan tersebut, diperlukan metode yang dapat menyelesaikan *timetable* dengan cepat dan presisi.

Telah banyak upaya peneliti untuk mendesain perangkat lunak yang dapat menciptakan *timetable* secara otomatis. Tetapi, sampai saat ini belum ada yang secara penuh menghasilkan solusi yang optimal. Selain itu juga masih belum memungkinkan untuk menciptakan *timetable* secara otomatis dimana semua kendala terselesaikan dengan baik. (Kumar, Sharma, & Mehta, 2012).

Ada beberapa variasi metode dari tujuan pemecahan masalah dalam pembuatan *timetable* dan pembuatan *schedule* (Norberciak, 2008). Variasi ini dibagi menjadi empat tipe :

- Metode Sekuensial**, metode ini memberi perintah kepada *event* dengan menggunakan domain heuristik dan memasukkan *event* secara berurutan ke dalam periode waktu yang valid atau disebut juga *timeslot* dimana tidak ada *event* dalam setiap periode yang saling bermasalah.
- Metode cluster**, pada metode ini event masuk ke dalam cluster dimana setiap dua event masuk ke dalam cluster khusus. Kekurangan utama pendekatan ini ialah cluster dari event terbentuk dari awal dan bersifat fix / tetap dan akan menghasilkan kualitas *timetable* yang kurang baik.
- Metode berbasis kendala**, Metode berbasis kendala, pada metode ini, kendala yang ada terbentuk dari beberapa variabel domain yang terbatas tergantung dengan sumber daya seperti periode waktu dimana *timetable* ditetapkan untuk memenuhi sejumlah kendala. Pada metode berbasis kendala, dapat terjadi dimana tidak semua kendala dapat terpenuhi.

- d) **Metode Metaheuristik**, variasi dari pendekatan metaheuristik untuk menyelesaikan timetable yaitu, Simulated Annealing (SA), Tabu Search, Evolutionary Algorithm, dan pendekatan lain. Metode metaheuristik dimulai dengan satu atau lebih solusi awal dan akan melakukan strategi pencarian untuk mendapatkan solusi yang optimal. Metode ini juga berupaya untuk menghindari localoptima. Pada masalah timetable yang lebih spesifik seperti pada perkuliahan, ada beberapa sesi pada satu hari.

Pada masalah *timetable* yang lebih spesifik seperti pada perkuliahan, ada beberapa sesi pada satu hari. Ada beberapa hari kuliah, yang berarti ada cukup banyak baris dan kolom tabel. Jika dalam sehari ada empat sesi dan dalam seminggu ada lima hari kuliah, sedangkan kampus memiliki empat 6 ruang kuliah, maka ada delapan puluh kolom yang akan terbentuk dalam tabel dari timetable perkuliahan.

Metode-metode yang pernah digunakan untuk menyelesaikan timetable adalah Traditional Forward Checking, Algoritma Genetik (GA), PSO (Chu & Chen, 2006), Metaheuristic(Gani, 2004). Traditional Forward Checking membutuhkan waktu hitungan jam untuk menyelesaikan timetable (Montero, 2011) Pada masalah nyata, metode PSO dapat menyelesaikannya dalam waktu yang cepat(Montero, 2011).

Dalam penerapan PSO untuk masalah timetable universitas, urutan pemilihan mata kuliah dianggap sebagai partikel, dan tabel mata kuliah dapat dianggap sebagai suatu ruang waktu yang terdiri atas beberapa dimensi. Tujuan utama pada perhitungan untuk mencari timetable yang optimal ialah untuk meminimalkan pelanggaran terhadap kendala.

Constraints

Berdasarkan jenisnya, ada 2 jenis kendala, yaitu kendala yang bersifat tegas (hard constraints) dan kendala yang bersifat lemah (soft constraint) (Gani, 2004)(Montero, 2011). Kendala yang bersifat tegas (*hard constraint*) harus dipenuhi, sehingga tidak boleh terjadi pelanggaran terhadap kendala ini. Sedangkan kendala yang bersifat lemah tidak harus dipenuhi. Kendala ini lebih rumit diselesaikan daripada yang bersifat tegas. (Chu dkk, 2006).

Permasalahan bervariasi dari satu institusi ke institusi lain, namun sebagian besar pada time constraints yang serupa.(Islam, T., Shahriar, Z., Perves, M.A. and Hasan, 2016). Selain itu untuk jenis Constraints secara pemenuhan prioritasnya, constraints dapat dibedakan menjadi 2 bagian, yaitu (Poole, L., David, Alan K., 2010):

Selain itu untuk jenis *Constraints* secara pemenuhan prioritasnya, *constraints* dapat dibedakan menjadi 2 bagian, yaitu (Poole, L., David, Alan K., 2010):

1. *Hard Constraints* : adalah persyaratan yang harus dipenuhi dan tidak boleh dilanggar dalam pembuatan penyelesaian masalah.
2. *Soft Constraints* : adalah persyaratan tambahan yang biasanya merupakan sebuah permintaan khusus, tanpa adanya konsekuensi fatal jika tidak terpenuhi. Menentukan Constraints

Menentukan Constraints

Untuk menentukan isi sebuah variabel yang hendak diisi, dapat dilakukan melalui dua cara yaitu(Poole, L., David, Alan K., 2010):

1. *Most Constrained Variable*

Penentuan Variable yang pertama diisi dan berikutnya dimulai dari variabel yang paling banyak mengandung *constraint*.

2. *Least Constrained Variable*

Penentuan variabel yang pertama diisi dan berikutnya dimulai dari variabel yang paling sedikit mengandung *constraint*.

Generate(Pembangkitan)

Permasalahan penjadwalan yang mengaitkan antara keinginan pribadi pekerja dengan ketersediaan jam kerja dalam organisasi dibahas dalam (Gunawan & Toba, 2016) mengusulkan agar setiap keinginan pribadi tersebut dijadikan sebagai hal yang dicari solusinya pada tahap terakhir pada saat jadwal

diverifikasi oleh tim dalam organisasi. Konsep yang ditawarkan dalam (Gunawan & Toba, 2016) dengan membuat modul-modul yang terpisah antara pembangkitan jadwal awal, perbaikan under dan overstaffing dalam shift jam kerja, dan kemudian verifikasi keinginan pribadi dari pekerja. Hal tersebut dianggap cocok untuk dilakukan dalam konteks jadwal pekerja dalam organisasi dalam jumlah pekerja cukup banyak karena dapat mengalokasikan jadwal pada pekerja lain.

Array

Array sebagai struktur data yang paling sederhana dapat berdimensi satu atau yang disebut dengan larik, serta dapat juga berdimensi dua atau yang disebut dengan matriks. Matriks merupakan array dua dimensi. Array adalah koneksi data yang tipenya sama, tersusun dalam bentuk arus berurutan dan jumlah elemen atau datanya tidak berubah sesuai deklarasi awal. Berdasarkan konvensi, indeks pertama adalah baris, indeks kedua adalah kolom. (Putri, 2014).

Kesimpulan

1. Berdasarkan hasil survey pendahuluan yang dilakukan dalam sistem penjadwalan dan formulir kartu rencana studi yang diterima oleh mahasiswa dan program studi teridentifikasi beberapa permasalahan, antara lain :
 - a) Dalam antrian terjadi penumpukan antrian yang tinggi sehingga terjadi penumpukan user yang berakibatkan kinerja sistem menjadi loading yang tinggi dan berujung pada hang dan bahkan crashnya sistem tersebut.
 - b) Generate yang telah dilaksanakan sebaiknya tidak dilakukan proses batal tambah kembali.
 - c) Adanya Rencana penambahan slot pada penjadwalan yang terjadi semula 15 slot mengalami perubahan menjadi 17 slot.
2. Dengan metode Auto Generate Time Table dapat meningkatkan ketelitian dalam melakukan sistem penjadwalan yang sesuai dengan tabel perencanaan yang dilakukan bagian penjadwalan.
3. Hasil penjadwalan yang dilakukan menggunakan standar STMIK Raharja, sedangkan lainnya dapat diimplementasikan di kampus lain yang disesuaikan kebutuhan.
4. Hasil implementasi sistem penjadwalan yang dilakukan pada sistem online jadwal rencana semester (OJRS) memudahkan bagian administrasi pada divisi akademik dalam mempersiapkan rencana perkuliahan di awal semester.
5. Dengan demikian mahasiswapun dapat melakukan rencana perkuliahan dengan mengisi form Kartu Studi Tetap (KST) dan selanjutnya melakukan proses batal tambah diakiri dengan terbitnya Kartu Studi Tetap Final (KSTF).

Ucapan terima kasih

Penelitian ini dapat berlangsung baik karena dibiayai oleh skim Penelitian Dosen Pemula (PDP) Direktorat Penelitian dan Pengabdian kepada Masyarakat (DP2M) – DIKTI anggaran tahun 2018, oleh karena itu kami mengucapkan terima kasih kepada DP2M - DIKTI mengenai pendanaan penelitian.

DAFTAR PUSTAKA

1. Ariani, D., Fahriza, A., & Prasetyaningrum, I. (2011). Optimasi Penjadwalan Mata Kuliah di Jurusan Teknik Informatika Pens dengan Menggunakan Algoritma Particle Swarm Optimization (PSO). *EEPIS Repository*, 1–11.
2. Aziz, A. (2015). Optimasi Penjadwalan Perkuliahan Menggunakan Metode Simulated Annealing (Studi Kasus: Program Studi Teknik Informatika Universitas Yudharta Pasuruan). *EXPLORE IT : Jurnal Keilmuan Dan Aplikasi Teknik Informatika*, 7(2).
3. Chu, S., & Chen, Y. (2006). Timetable Scheduling Using Particle Swarm Optimization. *IEEE Xplore*,

- 0–3. <https://doi.org/10.1109/ICICIC.2006.541>
4. Gani, T. A. (2004). Optimizing examination timetabling using a hybrid evolution strategies Optimizing Examination Timetabling using a Hybrid Evolution Strategies. *2nd International Conference on Autonomous Robots and Agents*, (June 2017), 345–349.
 5. Gunawan, C. A., & Toba, H. (2016). Pembangkitan Solusi Penjadwalan Berprioritas Melalui Penerapan Constraint Satisfaction Problem (Studi Studi Kasus : Laboratorium Fakultas Teknologi Informasi Universitas XXX). *Jurnal Teknik Informatika Dan Sistem Informasi*, 2(April), 43–52.
 6. Islam, T., Shahriar, Z., Perves, M.A. and Hasan, M. (2016). University Timetable Generator Using Tabu Search. *Journal of Computer and Communications*, 4, 28–37. <https://doi.org/10.4236/jcc.2016.416003>
 7. Kumar, K., Sharma, R., & Mehta, K. (2012). Genetic Algorithm Approach to Automate University Timetable, 1(1).
 8. Montero, E. (2011). A PSO algorithm to solve a Real Course + Exam Timetabling Problem. *International Conference on Swarm Intelligence*, 2, 1–9.
 9. Norberciak, M. (2008). Universal Method for Timetable Construction based on Evolutionary Approach. *International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, 2(3), 174–179.
 10. Poole, L., David, Alan K., M. (2010). *Artificial Intelligence : Foundations of Computational Agents*. UK: Cambridge University Press. Retrieved from www.cambridge.org/9780521519007
 11. Pressman, R. S. (2010). *Software Engineering : A Precititioner's Approach*. (F. M. Schilling, Ed.) (7th ed.). New York: McGraw-Hill.
 12. Putri, Y. (2014). Pemanfaatan Matriks Jarang dengan Metode Algoritma Genetika Menggunakan Program Pascal. *Jurnal Matematika UNAND*, 3(1), 98–106.
 13. Siswono, T. (2013). Kombinasi Algoritma Genetika dengan Algoritma Palgunadi untuk Penjadwalan Mata Kuliah di Universitas Sebelas Maret. *ITSMart*, 2(2), 7–12. <https://doi.org/http://dx.doi.org/10.20961/its.v2i2.624>
 14. Warsito, A. B., & Yusup, M. (2014). Kajian Yii Framework dalam Pengembangan Website Perguruan Tinggi. *CCIT Journal*, 7(40), 437–451. Retrieved from <http://raharja.ac.id/acid/karyailmiah/karyailmiah/view/id/7030814>
 15. Yudihartanti, Y., Syukur, A., & Wahono, R. S. (2011). Analisis Komparasi Metode Mamdani dan Sugeno dalam Penjadwalan Mata Kuliah. *Jurnal Teknologi Informasi*, 7(2), 109–116.